LINEAR PREDICTION ANALYSIS OF CERTAIN V-CV UTTERANCES OF HINDI

by

Joshipura Bhushit Pradyumna

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1995

Linear Prediction Analysis of Certain V-CV Utterances of Hindi

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Master of Technology

by
Joshipura Bhushit Pradyumna

to the

Department of Electrical Engineering

Indian Institute of Technology, Kanpur

April, 1995

Name of Student Joshipura Bhushit Pradyumna Roll No 9310418

Degree for which submitted M Tech Department Electrical Engineering

Thesis Title Linear Prediction Analysis of Certain V-CV Utterances of Hindi

Name of Thesis Supervisor Dr S K Mullick

Month and Year of Thesis Submission April, 1995

Abstract

The linear prediction (LP) analysis yielding PARCOR coefficients for V CV utterances for a fixed V /ə/ is carried out for the first 25 consonants of Hindi alphabet The area functions calculated thereby are compared with the articulatory phonetic classification of the consonants The work also includes implementation of the SIFT algorithm for pitch determination and corresponding speech synthesis It is concluded that the traditional LP model of speech analysis is unable to capture the articulatory phonetic pattern of Hindi alphabet

Certificate

This is to certify that the work contained in this thesis, entitled **Linear Prediction Analysis of Certain V-CV Utterances of Hindi**, has been carried out by **Joshipura Bhushit Pradyumna** under my supervision and that this work has not been submitted elsewhere for a degree

April 17, 1995

Dr S K Mullick
Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Kanpur, India

Acknowledgement

I thank Dr S K Mullick, a mentor with unique insight into the processes going in the mind and the heart of the student as well as the signal processing, to give me opportunity to learn research in its real spirit I thank him for his continuous attention, constant encouragement and assiduous guidance which has helped me to change the direction of my career This is a proper place but too little a space to express my gratitude

I also thank Dr Mrs Achala Raina, Dr P K Kalra and Dr G C Ray for various helps I received from them I also thank Dr D K Jha for introducing me to the grammar and phonetics of Sankrt

A number of friends helped me during my stay here In thesis matters I should not forget to thank Puranjay, Madhusudan, Madhav, Sudhir and Amitabh for helping me and spurring me from time to time Joydeep the real robust structure of friendship from structure engineering I thank him for bearing all the emotional load of my stay here My special thanks to S K Joshi and Nimisha*bhabhi* for being an oasis in this campus

Bhushit

(or *jbp* or even *J B Pradyumna* for this campus !)

Contents

List of Abbreviations

The following abbreviations are followed in the report

Abbreviation	Meaning
r v	random variable
m s	mean square
V	Voiced
UV	Unvoiced
A	Aspirated
UA	Unaspirated

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

Linear Prediction has been a popular tool for speech processing since its introduction to the field There are several advantages and disadvantages associated with this method as a speech processing tool [1] Still because of its acccuracy and simplicity it is one of the most popular techniques

Among various representations of the results of linear prediction analysis, reflection coefficients (PARCOR coefficients) are more interesting This is because of some properties of the lattice filter associated with it e g orthogonality, low coefficient sensitivity etc Above all, since the reflection coefficient is a normalized quantity, its magnitude is always bounded [2] It is empirically established that for telephonic quality speech, consideration of first ten reflection coefficient is sufficient [3,4]

Reflection coefficients are related to a number of other sets of parameters [1] Examples of such parameters are the log area ratio and the area function

A vocal tract can be modelled as a series of cylindrical waveguides composed of the same length but different diameters i^{th} log area ratio is nothing but the natural logarithm of ratio of area of $(i+1)^{st}$ cylindrical section to that of i^{th} cylindrical section. i^{th} area function is $(p-i)^{th}$ area measured in terms of glottis area (A_p) normalised to unity.

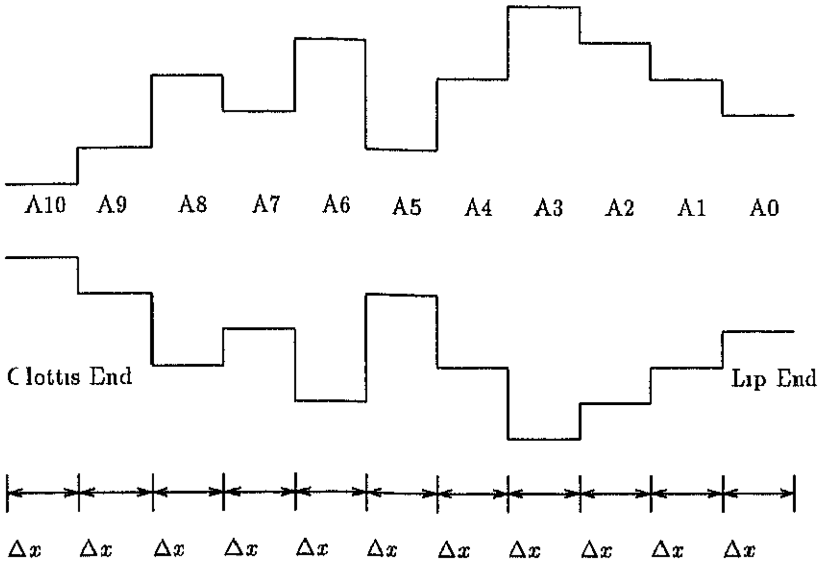


Figure 1.1: Wave Guide Model of Vocal Tract

There is an isomorphism between the i^{th} reflection coefficient and i^{th} log area ratio. That results into a recursive relation between area parameters and reflection coefficients. This is the first point of motivation to this work.

Let us consider phonetics for a while. Most of the Indian languages show a regular articulatory phonetic pattern in their alphabet(s). Every non-vowel character represents a CV utterance with a fix V /ə/. The first 25 non-vowel characters are classified in a startlingly systematic articulatory phonetic order [5,6,7]. The classification according to modern phonology [8] is given in table 1.1.

Place of Aıtıcultıon	Mannei of Aıtıculatıon				
	Unvoıced		Voıced		Nasal
	Unaspııated	Aspııated	Unaspıı ated	Aspııated	
Velaı	(क) /kə/	(ख) /k^{h}ə/	(ग) /gə/	(घ) /g^{h}ə/	(ङ) /ŋə/
Alveolaı	(च) /t ʃə/	(छ) /t ʃ^{h}ə/	(ज) /dʒ ə/	(झ) /dʒ^{h}ə/	(ञ) /ɲə/
Retıoflex	(ट) /ʈə/	(ठ) /ʈ^{h}ə/	(ड) /ɖə/	(ढ) /ɖ^{h}ə/	(ण) /ɳə/
Dental	(त) /tə/	(थ) /θə/	(द) /də/	(ध) /ðə/	(न) /nə/
Bılıbııl	(प) /pə/	(फ) /p^{h}ə/	(ब) /bə/	(भ) /b^{h}ə/	(म) /mə/

Tıble 1 1 Fıist 25 CV clusteıs of Hındı alphabet

Tıble 1 1 wıll hencefoıth be ıefeııed to as *Alphabet matrıı* Eıch ıow of the ılphabet matıix coııesponds to a sıngle place of ııtıculatıon With thıs ınd the ıecuısıon stıted above, one would expect to see some sımılaıity ın the tıajectoııes of aıea functıons ıow wıse (The alveolır non nasals aıe affıicates, all otheı non nasals aıe stops)

The compaıison of analysıs thıough the waveguıde model and the aıtıculatoıy phonetıc behavıouı of a V CV clusteı becomes moıe ınteıestıng because of the followıng facts

- The waveguıde model assumes constant vocal tube dıameteıs oveı a gıven analysıs fıame The classıficatıon of consonants ın aıtıculatoıy phonetıcs ıs based on vaııous constııctıons ın the vocal tıact Thıs ınvolves dynamıc behavıouı of the vocal tıact

- In the wıveguıde model, all the sectıons of the vocal tıact aıe assumed to have ıdentıcal length The consonants on the otheı hand, aıe classıfied as peı constrıctıons at places nonunıfoımally dıstııbuted oveı the vocıl tıact

Alternately stated, comparison of reflection coefficient trajectories should reflect the validity as well as some possible shortcomings of the cylindrical waveguide model

It is with this consideration in mind that an attempt is made in this investigation to characterize the sounds corresponding to Devanagari alphabet (Vyanjana section) using LPC model and study its usefulness for text to speech conversion for Indian languages Another stronger point for motivation is the availability of hardware products dedicated to linear prediction analysis [9]

To study the trajectories of the parameters the recording is carried out using 'Speech Interface Unit' [10] More details of the recording can be found in Appendix A

As a part of the work a software is developed which is briefly described below

1 Analysis section This section derives the gain and reflection coefficient s through Durbin's algorithm [1] and the pitch through SIFT algorithm [11]

2 Synthesis section This section utilizes a two multiplier lattice structure Its main purpose is to see the validity of the results of the analysis section

3 Conversion section This section consists of various programmes written in C to convert the data formats into ones required by various supporting softwares available ~~in the campus~~

4 Control section This section consists of a standard K shell UNIX makefile, a programme written in C which takes in an argument and executes various programmes by supplying them the argument with appropriate augmentations and attributes and another C file containing a function which terminates the calling programme if the file pointer passed to it is NULL

The control section may seem trivial at present but will be extremely useful for organization of database while developing full scale text to speech conversion system In order to maintain the portability of the C code ANSI C standard is followed throughout

The organization of thesis is as follows

- Chapter 2 discusses organization of Hindi/Sankrt alphabet according to the modern phonology

- Chapter 3 discusses some aspects of linear prediction, synthesis model, SIFT algorithm for pitch determination and log area ratios

- Chapter 4 overviews the software developed

- Chapter 5 discusses the results

- Chapter 6 covers conclusions and suggestions for future work

- Appendix A gives details about the recording of database

Chapter 2

Organization of Hindi Alphabet

Like most of the other Indian languages, Hindi too has an alphabet of phonetic nature It has two distinct parts vowels,dipthongs and some trills (स्वर) /svərə/ and CV clusters with a fixed V /ə/ The latter section is called (व्यंजन) /vjəɲdʒ ə nə/ For convenience, this section will be called *CV clusters*

Pure consonants are represented by a diacritical mark '्' called (हलन्त) /hələntə/ CCV and CCCV clusters are represented by joined orthographical representation [5,6]

2 1 Vowels, Dipthongs and Some Trills

This section has members in the following order (अ) /ə/, (आ) /a/ (इ) /ı/, (ई) /ı /, (उ) /u/, (ऊ) /u /, (ऋ) /ı/, (ॠ) /ı /, (ऌ) /l^r/, (ए) /e/, (ऐ) /aı/, (ओ) /o/, (औ) /au/, (अ) /ã/ and (अ) /a^h/

2.2 CV Clusters with a fix V /ə/

Hindi is affluent in consonants. Hindi is a Sanskrit based language. Sanskrit grammarians rejected possibility of utterance of consonants without a vowel following or preceding it [5,6]. So Sankrt and hence Hindi have characters representing CV utterences. For a constant C, such a collection is called (स्वरमाला) /svarəmala/ and is considered to be a derivative of the alphabet which is nothing but the collection of first letter of /svarəmala/. So we get a constant vowel /ə/ in Hindi alphabet.

- For first 25 characters we get a periodic pattern of UV UA, UV A, V UA, V A and nasal CV clusters. Columnwise they are velar stops, alveo palatal affricates, retroflex stops, dental stops and bilabial stops (except for nasals of course, which are resonants) respectively [Table 1.1].

- The other section contains the following CV clusters in the following order (य) /jə/ palatal glide, (र) /rə/ retroflex trill (ल) /və/ alveolar lateral, (व) /lə/ dental glide, UV fricatives of palatal retroflex and dental articulatory positions (श) /ʃə/, (ष) /ʂə/ and (स) /sə/, and (ह) /hə/ V velar fricative.

2.3 Further Notes on Orthography and Phonetics of Hindi

In spite of this vast collection of consonants, this covers neither full Devnagari alphabet nor a complete set of Hindi phonetics. To take care of Persian and English influences and dialectial requirements, Hindi has adopted various diacritical marks (e g ‘ ’ (नुक्ता) /nuqta/ is used to represent ‘back consonants)

Still some other established CV clusters in one or the other Indian languages are not used in Hindi (e g (ళ) /lə/) Hindi or any other Indian alphabet is not able to cover allophonic or co articulatory effects

2 4 Choice of the Basis of Comparison

In addition to the above facts and likely systematic behaviour of parameters,phonetically there is not much discrepancy in the first 25 CV clusters among most of the Indian languages

In order to see the variation of parameters over the alphabet matrix, a 10 coefficient lattice structure is chosen With this overview of Hindi alphabet and its ordering, it seems important to see how the aformentioned model behaves with respect to different CV clusters

Since the initial errors in linear prediction model are large and the consonant duration and energy in a CV cluster are (generally) less than that of vowel counterpart, it is not meaningful to choose CV utterance as the base of comparisons This conjecture was also checked practically by synthesizing on the developed software and available setup A lot of ambiguity and distortion were detected So the analysis was carried out on V CV clusters

Chapter 3

The Linear Prediction

The motivation behind choosing a linear prediction model was mentioned in the Chapter 1. For the sake of completion, this chapter briefly discusses its well known theory and the SIFT algorithm for pitch determination.

3.1 Linear Estimation

General problem of linear estimation can be stated as follows.

We are given p random variables (r.v.s) $\{\mathbf{X}_i\}_{i=1}^{i=p}$ as *data* and we wish to find p constants $\{a_i\}_{i=1}^{i=p}$ such that if we estimate the r.v. s (*signal*) by the sum

$$\mathbf{s} = \sum_{i=1}^{p} a_i \mathbf{X}_i \tag{3.1}$$

so that the m.s. value $E\{\mathbf{e}^2\}$ is minimum, where,

$$\mathbf{e} = \mathbf{s} - \mathbf{s} \tag{3.2}$$

$$\mathbf{e} = \mathbf{s} - \left(\sum_{i=1}^{p} a_i \mathbf{X}_i \right) \tag{3.3}$$

s becomes the homogeneous linear MS estimate of the signal in terms of data. This estimate is given by the conditional mean $E\left\{\mathrm{s} \mid \{\mathbf{X}_i\}_{i=1}^{i=p}\right\}$

As per projection theorem, $E\{\mathrm{e}^2\}$ is minimum for $\{a_i\}_{i=1}^{i=p}$ if the error e is orthogonal to the data.

$$E\{\mathrm{e}\mathbf{X}^*\} = 0 \qquad 1 \le i \le p \tag{3.4}$$

for real $\mathbf{X}_i$s,

$$E\,\{\mathrm{e}\mathbf{X}_i\} = 0 \qquad 1 \le i \le p \tag{3.5}$$

$$\Rightarrow E\left\{\left(\mathrm{s} - \sum_{j=1}^{p} a_j\mathbf{X}_j\right)\mathbf{X}_i\right\} = 0 \qquad 1 \le i \le p \tag{3.6}$$

Setting $i = 1, 2, \ldots, p$, $R_{ij} = E\{\mathbf{X}_i\mathbf{X}_j\}$ and $R_{0j} = E\{\mathrm{s}\mathbf{X}_j\}$ we get the solution

$$\sum_{i=1}^{p} R_{ij}a_i = R_{0j} \qquad 1 \le j \le p \tag{3.7}$$

These are the well known *Yule Walker equations*.

3.2 Linear Prediction of Discrete Time Series in terms of Finite Past

The r step predictor of a discrete random process $\mathrm{s}[n]$ is the estimate of $\mathrm{s}[n+r]$ in terms of $\mathrm{s}[n]$ and its past.

$$\mathrm{s}[n+r] = E\{\mathrm{s}[n+r] \mid \mathrm{s}[n-k],\ k \ge 0\} \tag{3.8}$$

For 1 step predictor and stationary process s,

$$s[n] = E\{s[n] \mid s[n-k],\ k \geq 1\} \tag{3 9}$$

$$s[n] = \sum_{k=1}^{\infty} a_k s[n-k] \tag{3 10}$$

However for practical consideration, we limit ourselves to at most p past values of s[n],

$$s_p[n] = E\{s[n] \mid s[n-k],\ 1 \leq k \leq p\} \tag{3 11}$$

$$s_p[n] = \sum_{k=1}^{p} a_k s[n-k] \tag{3 12}$$

so the forward error becomes

$$e_p[n] = s[n] - s_p[n] \tag{3 13}$$

from the projection theorem,

$$E\{(s[n] - s_p[n])s[n-k]\} = 0, \qquad 1 \leq k \leq p \tag{3 14}$$

and thus we get the Yule Walker equations in the form

$$R[1]a_1^p + R[2]a_2^p + \quad + R[p]a_p^p = R[0] - \frac{\Delta_{p+1}}{\Delta_p}$$

$$R[0]a_1^p + R[2]a_2^p + \quad + R[p]a_p^p = R[1]$$

$$R[p-1]a_1^p + R[p-2]a_2^p + \quad + R[0]a_p^p = R[p]$$

where Δ_p is the determinant of the correlation matrix R_p with the co efficients of the last p equations We note here that the elements on each diagonal of the matrix of R_p are identical i.e., the matrix is *Toeplitz*. Hence we can apply numerous recursive algorithms to solve for $\{a_p^k\}_{k=1}^{k=p}$

3.3 Durbin's Algorithm

This algorithm is one of the most efficient recursive algorithm to solve the above problem. Throughout the discussion in this section $e[i]$ denotes error at the i^th step of the lattice filter

```
BEGIN
e[0] = $R[0]$,
 FOR $i = 1$ TO $i = p$ DO
    BEGIN
    $k[i] = \left(R[i] - \sum_{j=1}^{i-1} \alpha_j[i-1]R[i-j]\right) / e[i-j]$,
    $\alpha_i[i] = k[i]$,
     FOR $j = 1$ TO $j = (i-1)$ DO
        BEGIN
        $\alpha_j[i] = \alpha_j[i-1] - k[i]\alpha_{i-j}[i-1]$,
        END,
    $e[i] = (1 - k^2[i])e[i-1]$,
    END,
END
```

The final solution is given by

$$a_k^p = \alpha_k^p, \ 1 \leq k \leq p, \tag{3.15}$$

and

$$E\{e^2\} = e[p] \tag{3.16}$$

3.4 The Synthesis Model

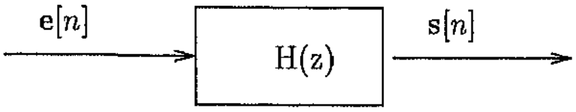


Figure 3.1 All Pole Filter

Let $H(z)$ be an all pole filter with the transfer function

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{3.17}$$

and $s[n]$ be the output of the filter for some input process $e[n]$ i.e.,

$$s[n] = Ge[n] + \sum_{k=1}^{p} a_k s[n-k] \tag{3.18}$$

The solution to the determination of $\{a_k\}_{k=1}^{p}$ such that $E\{e^2\}$ is minimized is discussed in previous sections. Now comes the question of determination of the nature of $e[n]$.

In unvoiced sections, speech signal s[n] is noiselike. So, selection of a random noise generator is proper as the source $e[n]$. On the other hand, in voiced sections, speech signal is almost periodic: a damped sinusoid. So one would use as excitation, a periodic impulse generator with the period being the pitch period In either of the cases, the m s value of the signal e$[n]$ and the filter coefficients

Any IIR/FIR system with rational transfer function can be represented by direct implementation form or its lattice equivalent In our case the direct implementation coefficients are $\{a_k^i\}_{k=1}^{k=p}$ The lattice coefficient counterpart $\{k[i]\}_{i=1}^{i=p}$ is obtained through Durbin's algorithm[1] We can utilize the IIR lattice structure with reflection coefficients, a gain block, a V/UV switch, a random number generator and a periodic impulse generator (with controllable period) to form the synthesis model

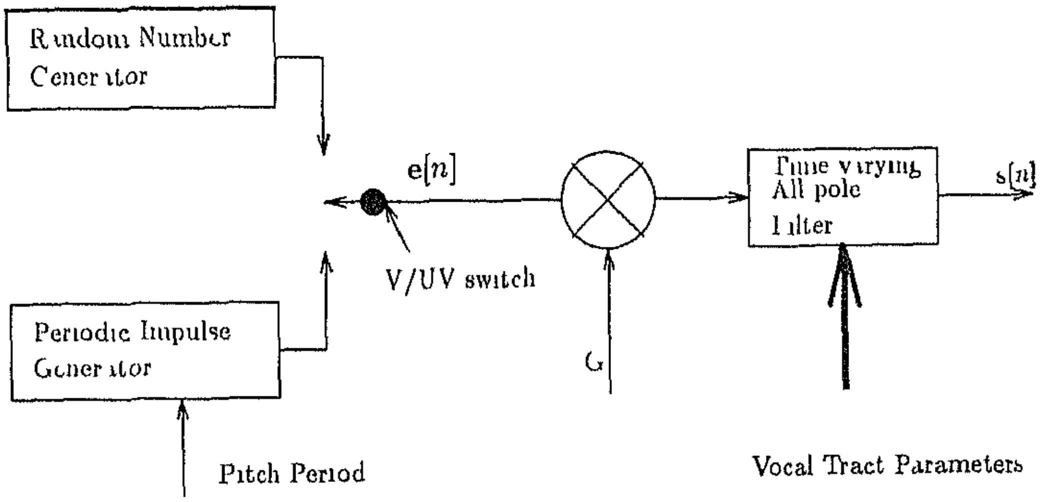


Figure 3 2 The Synthesis Model

3 5 V/UV decision and Pitch Determination the SIFT algorithm

There are a number of algorithms available for both V/UV decision and pitch determination[1,2,12] SIFT algorithm is selected bacause it is moderately accurate and computationally shares some functions with the linear prediction analysis

Idea of SIFT algorithm is to pass the given speech through a low pass filter to remove higher formants, then to pass the output thorgh low order inverse filter, (continuing with the notations of earlier sections,inverse filter is the one which takes s[n] as the input and gives e[n] as the output) and autocorrelate the output to determine whether it is voiced or not[2,10]

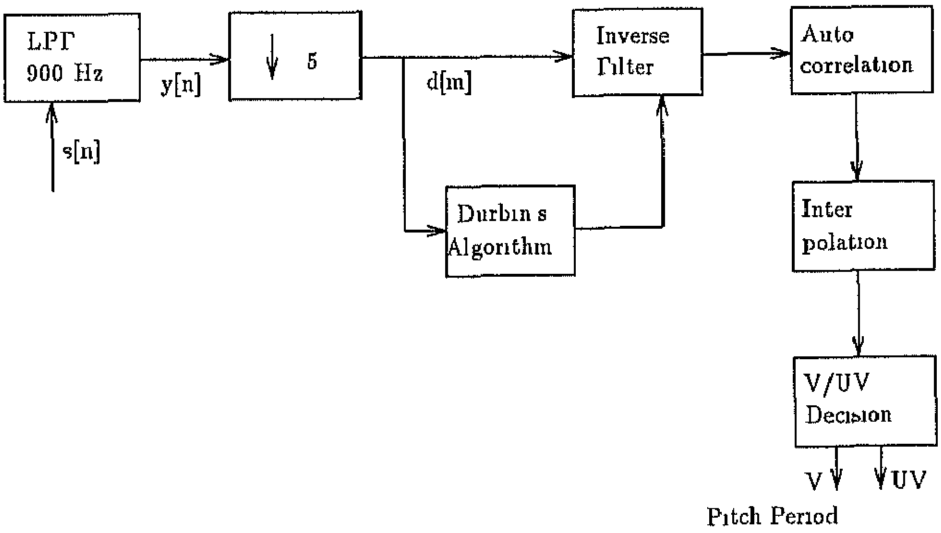


Figure 3 3 Block Diagram of SIFT Algorithm

Details of the implementation may be found in Chapter 4.

3.6 Relations with Cylindrical Waveguide Model

Though the theory behind the derivation of the relations between the reflection coefficients and the log area ratios is rather involved, the assumptions behind such a modelling and the relation by itself are noteworthy[2].

Assumptions

1. The vocal tract is assumed to consist of p inter connected sections of equal length. Each individual section is of uniform area.

2. The transverse dimension of each section is small enough compared with a wavelength so that the sound propagation through an individual section can be treated as a plane wave.

3. The sections are rigid so that the internal losses due to wall vibration, viscocity and heat conduction are negligible.

4. Normal assumptions leading to elementary wave propagation are valid.

5. The model is linear and is uncoupled from glottis.

6. The effects of the nasal tract can be ignored.

Relation

Let A_m be the cross sectional area of m^{th} section and g_m be the m^{th} log area ratio, then the relation is given by

$$g_m = log\left[\frac{A_{m-1}}{A_m}\right] = log\left[\frac{1+k_m}{1-k_m}\right] \qquad (3.19)$$

3 7 Selecting Various Parameters for the Model

1 Sampling Rate FS Since LPC based speech processing is meant for telephone quality speech with passband 300~3500 Hz the sampling frequency of 8000 Hz is chosen

2 Order of Filter P Empirically established value is 10 From vocal tract model, the model should be able to memorize upto $2L/c$ seconds, where L is the average length of the vocal tract ($\approx 17cm$) and c is the velocity of sound ($\approx 34cmS^{-1}$) Therefore necessary memory becomes 1 mS For 8 kHz sampling, it comes out to be 8 To be nearer to the standards P is kept to be 10

3 Analysis Frame Length N This should be small enough so that the vocal tract movement can be considered negligible ($\approx 25mS$) This comes out to be 200 For transient sounds, smaller interval is desirable

4 Windowing In order to avoid spectral abruptness because of the difference between **s**[0] and **s**[$n-1$] windowing is desirable So a Hamming window is applied

5 Synthesis Frame Length framelen In order to maintain smoothness in speech reproduction, 2/3 of the analysis length is taken to be the synthesis framelength i e , there is an overlap of 1/3 frame for the analysis

Chapter 4

Software Overview

As discussed in introduction, the software developed can be divided into four sections (i) Analysis section (ii) Synthesis section (iii) Conversion section and (iv) Control section

4 1 Analysis Section

There is a programme *lpcan c* which takes in a binary string file of 16 bit signed integer speech data and gives a binary string file of floating point values *of gain, pitch period and $k[0]$ to $k[10]$ the reflection coefficients* frame by frame (Of course $k[0]$ is a useless quantity and is taken care by the synthesis section) Important sections of the algorithm is discussed below

```
    Program lpcan c
    #include "lpc h",
    #include "lpcan h",
    array h[N],fa[6] of double,

BEGIN
integer n, frame,
Assign default values to the file pointers,
Modify file poiter values considering the command line, (* 'asgnargv' *)
Check the value of file pointers, (* 'checkNull' *)
Calculate the Hamming window h[ ], LPF coefficients fa[ ], (* globals *)
n  = 0, (* sample index *)
frame  = 1, (* frame index *)
REPEAT
        BEGIN
        IF (NOT (EOF(input file))) THEN
                BEGIN
                Read an integer from input file, (* speech sample *)
                Pass the sample through 900 Hz LPF, (* 'lpf1  *)
                Increment the sample index,
                END,
        IF ((n = N) OR (EOF(input file))) THEN
                  BEGIN
                  Calculate gain and reflection coefficients, (* 'calcCoeff' *)
                  Calculate pitchperiod, (* 'calcPitch' *)
                  Note down the coefficients in the output file,
                IF (verbose mode) THEN display frame # and parameters,
                Shift the last $\frac{1}{3}$ part of frame as the first $\frac{1}{3}$ part and clear
                        the rest of the frame,
                Modify the sample index and frame index,
                END,
        END,
        UNTIL (EOF(input file)),
        Close the files,
END
```

function 'asgnargv checks for the arguments " sf", " pf" and " v" in the command line and assigns the argument next to them to input file pointer, output file pointer and verbose flag The operation is carried out in a 'WHILE DO' loop and hence no order of argument specification is imposed If there is some mistake in the command line, it terminates the programme displaying 'usage' message at the file 'stderr'

function 'checkNull c' is the most referred function It takes in a file pointer and a string In case the file pointer points to NULL, it terminates the calling programme and displays the string in the file 'stderr'

function 'lpf1' is a second order LPF with cut off frequency corresponding to 900 Hz in analog domain

function calcCoeff(s[N] array of short integers, h[N], VAR k[P+1] array of float, VAR G float)

(* s[ ] is the speech frame, h[ ] the Hamming window, k[ ] reflection coeff array and G is the gain of the model *)

```
    w[N], r[P+1] : array of float,
    (* w[ ] is the 'Hammed' speech frame, r[ ] is autocorrelation array *),

BEGIN
window(s,h,w,N), (* window the speech *)
average(w,N), (* make the average of w[ ] zero *)
(* The Linear Prediction results are for 'homogeneous' estimates *)
autoCorrel(w,N,P,r), (* calculate first P+1 autocorrelations *)
durbin(r,P,k,G,P), (* Durbin's algorithm *)
END
```

function 'calcPitch' is the implementation of the SIFT algorithm discussed in Chapter 3. After the decimation, the average of the signal is made zero. The interpolation is parabolic with the values adjecent to the maximum of autocorrelation taken into account. The UV/V decision is taken by passing the interpolated maximum value 'rval' alongwith the interpolated possition 'x' to the function 'decision'

```
    function decision(rval, x, VAR per : float),
    static vuv : integer, (* flag to refer to the decision about the previous
frame *)

BEGIN
IF ((rval ≥ 0.4) OR ((rval ≥ 0.3) AND (vuv = 3)),
        BEGIN
        vuv := (vuv & 1) * 2 + 1, (* this frame is voiced *)
        per := x, (* return the pitchperiod *)
        END
ELSE
      BEGIN
      vuv = (vuv & 1) * 2, (* this frame is unvoiced *)
      per = 0.0, (* return the pitchperiod showing this *)
      END,
END
```

4.2 Synthesis Section

```
Program lpcsyn.c
   #include "lpc.h",
   #include "lpcsyn.h",

BEGIN
integer frame,
Assign default values to the file pointers,
Modify file pointer values considering the command line, (* 'asgnargv' *)
Check the value of file pointers, (* 'checkNull' *)
frame = 1,
Allocate memory to the synthesis buffer,
IF (no allocation of memory) THEN
      BEGIN
      Display error message,
      Exit,
        END,
    REPEAT
            BEGIN
            Read in the frame of the parameters,
            IF ( verbose mode ) THEN display frame # and parameters,
            Synthesize the speech using the parameters, (* 'synthesize' *)
            IF (frame $\neq$ 1) THEN store the speech,
            Increment frame index,
            END,
    UNTIL (EOF (input file)),
    Close all the files,
    Free the memory occupied by the synthesis buffer,
    END
```

```
    function synthesize (frame   integer, gain, k[P+1], VAR y[framelen]
float,)

BEGIN
IF (frame = 1) THEN
            BEGIN
            Initialize all the 'previous' parameters by present parameters,
            Return,
            END
ELSE
      BEGIN
    Set present parameters into local static variables as new parameters,
    IF (both the previous and the present frames are voiced) THEN
            set slopes for all the parameters for linear interpolation,
    ELSE set all slopes to zero,
    FOR i = 0 TO i = framelen DO
        BEGIN
        Synthesize using two multiplier lattice model and present
                parameters,
        Update present parameters using linear interpolation,
        END,
    Set old parameters equal to new parameters,
END
```

4.3 Conversion Section

This section consists of various executable files which convert data from one format to the other. This section is support dependant. This section should modified as and when required. This section also includes executable file to convert the reflection coefficients into log area ratios. All the executable files have their source code files with an extension ".c". The summary of programmes is as below.

- voice2sh.c : converts "Speech and Voice Systems" format file into binary string of signed short integers.

- ns.c converts the binary string file of short integers into 'gnuplot' compatible ASCII file

- log.c finds the reflection coefficients from the parameter files and converts them into log area ratios The output file is 'gnuplot' compatible

- spch.c cuts the file portion interactively asking from where the speech starts and where it ends

- sh2voice.c converts the binary string file of short integers into the "Speech and Voice Systems" data format file

- att2int.c converts 2's complement swapped bytes binary string files of short integers into binary string of signed short integers

- area.c finds the reflection coefficients from the parameter file, calculates area parameters and makes 'gnuplot' compatible files of area function as a function of time (2 Dimension plot) and the overall vocal tract shape as a function of time (3 Dimension plot)

4 4 Control Section

This section consists of a standard K shell UNIX makefile, a programme written in C which takes in an argument and executes various programmes by supplying them the argument with appropriate augmentations and attributes and another C file containing a function which terminates the calling programme if the file pointer passed to it is NULL

This section is useful to maintain uniformity of file naming and to reduce the processing effort Just by entering 'process' and the sound file name (with '.dat' extension by default) we get proper sequential execution of all the programmes in the sections discussed above. This is accomplished with a fixed fashion of augmentation of the sound file name for the name of file generated at a particular stage of processing.

4.5 Flexibility Aspect

In the major section of analysis and synthesis the external functions are divided into two C files: 'common.c' and 'tool.c'. 'common.c' contains argument assignment, file pointer null checking and usage functions. 'tool.c' contains general functions of digital signal processing like average filter, forward and inverse lattice filter, second order LPF, decimator, interpolator, autocorrelation, Durbin's algorithm etc.

No function of 'tool.c' takes any global variable for granted. So these functions can be added to the library of the general DSP functions and save a lot of development time. The development of such a small but useful toolbox was necessary for smooth implementation of the work.

All analysis and/or synthesis parameters (like framelengths, sampling rate, default file names, maximum and minimum values of pitch expected etc.) are defined in the header files. So study of variation in various parameters can be handled with a minimal effort.

Chapter 5

Discussion of Results

With 25 V C V clusters in consideration and 10 area functions for each of them, the actual graphical representation becomes volumenous Here a try is made to sketch the important results in brief

5 1 The Basis of Presentation

- Our interest is in the estimate of variation of area functions with respect to time for a given consonant – a transitory phenomenon The area functions are plotted linearly interpolated instead of discrete area functions as plotted in vowel analysis

- Since the interest is to see the reflection of articulatory position function as two the function of time, the function of articulatory position and the commonality for the same place of articulation, the dimension of the graph becomes unmanagable over a paper A solution to this is to see the area function of interest going columnwise in alphabet matrix For the sake of brevity, one representative column per the area parameter of interest is shown.

- The area functions of interest are those corresponding to articulatory positions. Such area functions (A_0 at lips, A_{10} at glottis, the latter normalized to unity) are as follows

Articulation	Area Function of Interest
Velar	A_6,A_5
Alveolar and Retroflex	A_3,A_2
Dental	A_2,A_1
Bilabial	A_1,A_0

Table 5.1 Place of Articulation and Area Functions of Interest

5.2 Conventions followed in Graphs

- The X axis has two lables: type of utterance and the area function. The type of utterance is UVUA: Unvoiced, unaspirated, UVA: Unvoiced, aspirated, VUA: Voiced, Unaspirated, VA: Voiced, Aspirated or Nasals. The area function A_i is represented as Ai.

- The Y axis shows the value of the area function with A_{10} normalized to unity. The plot is linearly interpolated.

- The curve for A_i involving a particular consonant C is named 'string Ai', where 'string' is a string dependent on C. Table 5.2 shows the string equivalent of each member of the alphabet matrix. The alveolar non nasals are affricates, all other non nasals are stops.

Place of Articulation	Manner of Articulation				
	Unvoiced		Voiced		Nasal
	Unaspirated	Aspirated	Unaspirated	Aspirated	
Velar	ak	akh	ag	agh	ang
Alveolar	ach	achh	aj	ajh	anj
Retroflex	at	athh	ad	adhh	ann
Dental	atden	ath	adden	adh	an
Bilabial	ap	aph	ab	abh	am

Table 5.2 Curve Prefixes Involving Various Consonants

5.3 Observations

The important observations from the results are summarized below.

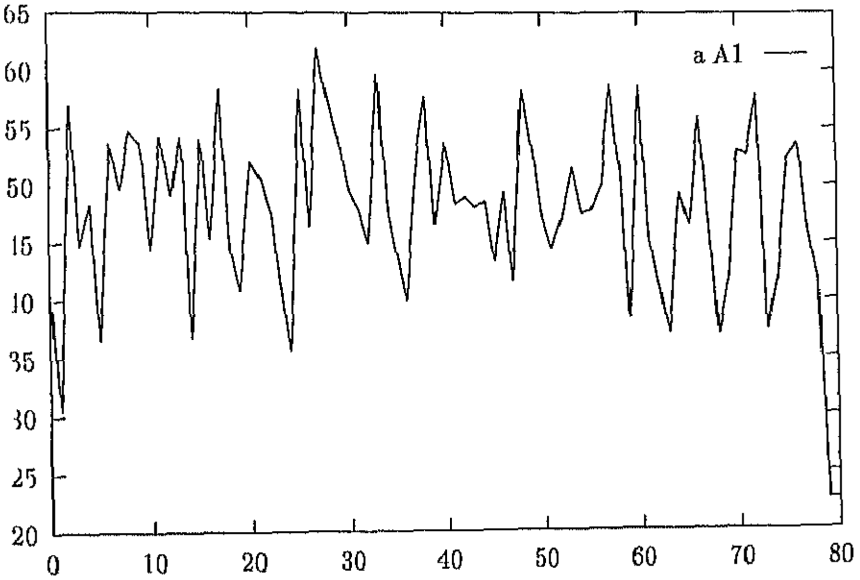
1. For a sustained vowel, the area functions show oscillatory behaviour [Fig. 5.1].

2. The silence threshold detection works well with almost all the utterances except some V CV utterances with voiced aspirated consonant.

3. It is easy to distinguish between the vowel part and consonant/silence part from the graphs. Vowel show more vocal opening than the consonant counterpart. Throughout the vocal tract, the consonant sections show construction.

4. In case of unaspirated consonants, voiced and unvoiced consonants show almost similar trajectories.

5. Going column wise in the alphabet matrix, it is not possible to detect any particular constriction for particular place of articulation.



1 igui c 5 1 Oscıllatıons obseıved foı a Sustaıned Vowel /ə/

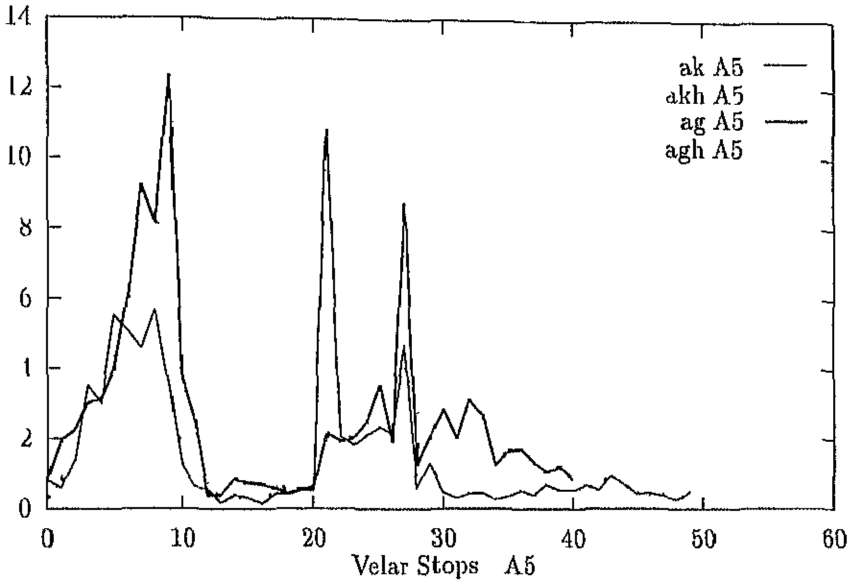


Figure 5 2 Parameter A_5 for Velar Stops

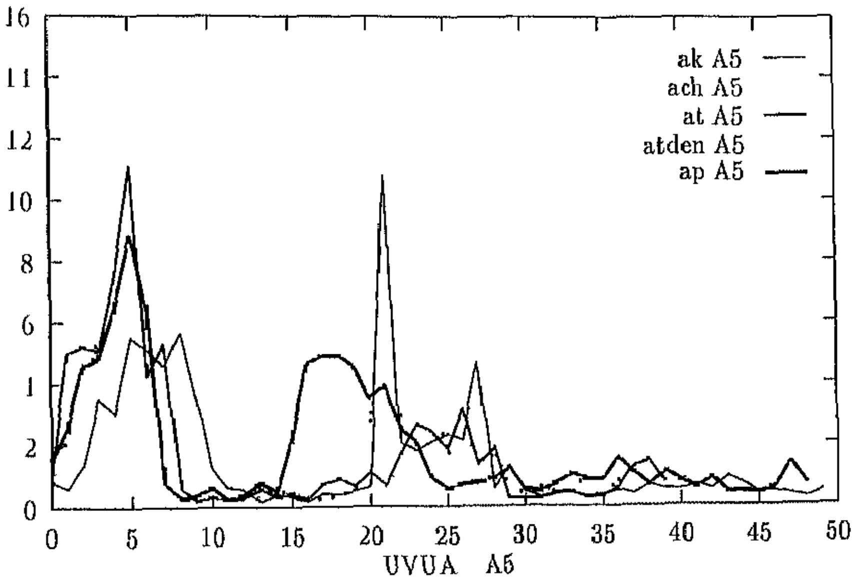


Figure 5 3 Parameter A_5 for Unvoiced Unaspirated Stops

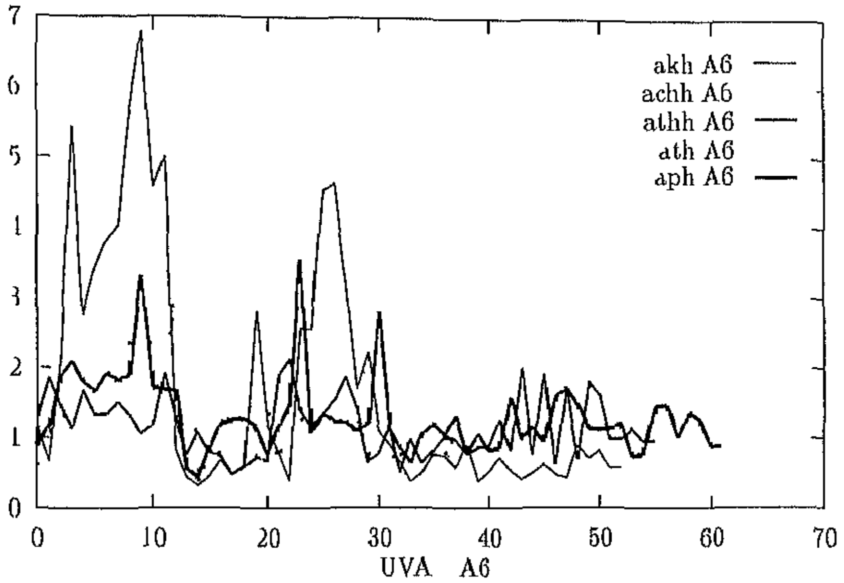


Figure 5 4 Parameter A_6 for Unvoiced Aspirated Stops

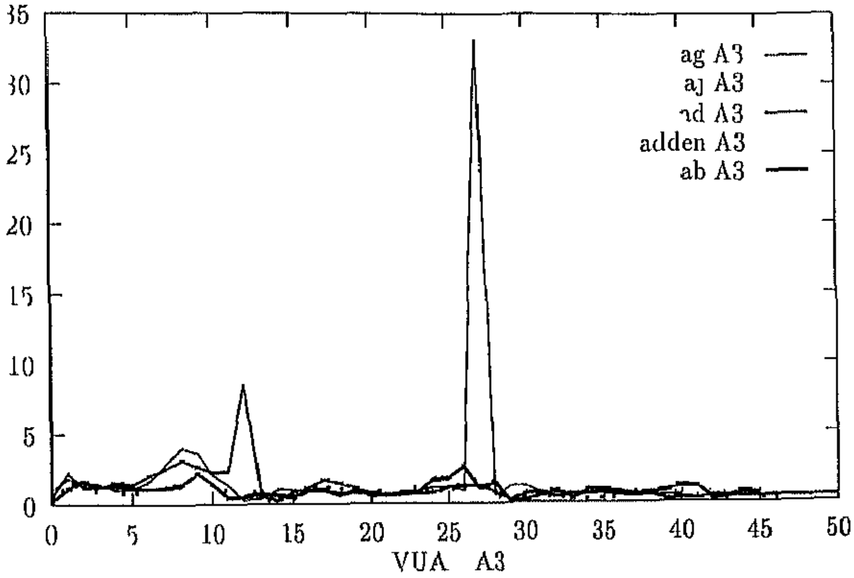


Figure 5.5 Parameter A_3 for Voiced Unaspirated Stops

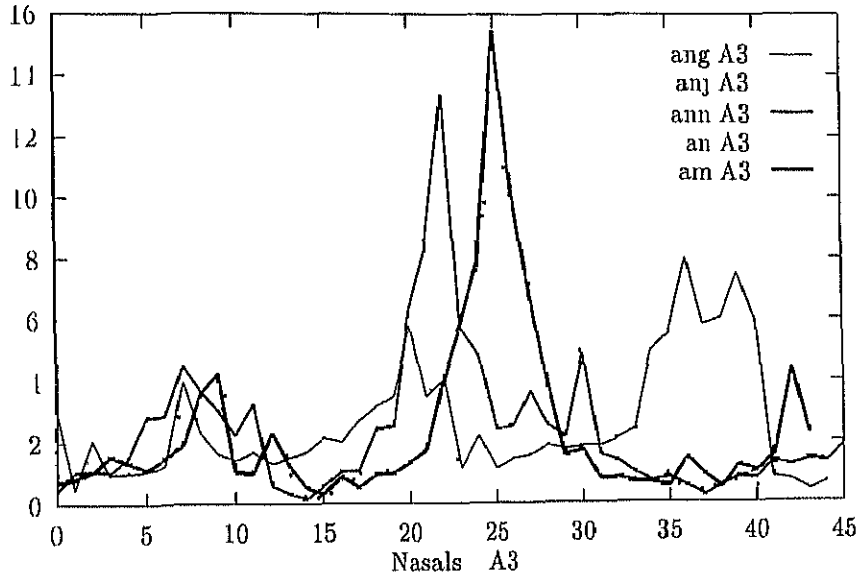


Figure 5 6 Parameter A_3 for Nasals

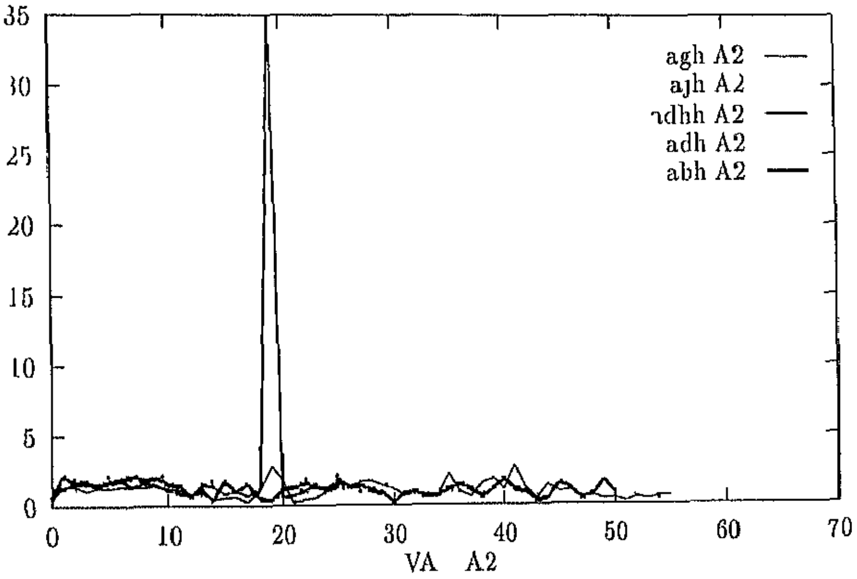


Figure 5 7 Parameter A_2 for Voiced Aspirated Stops

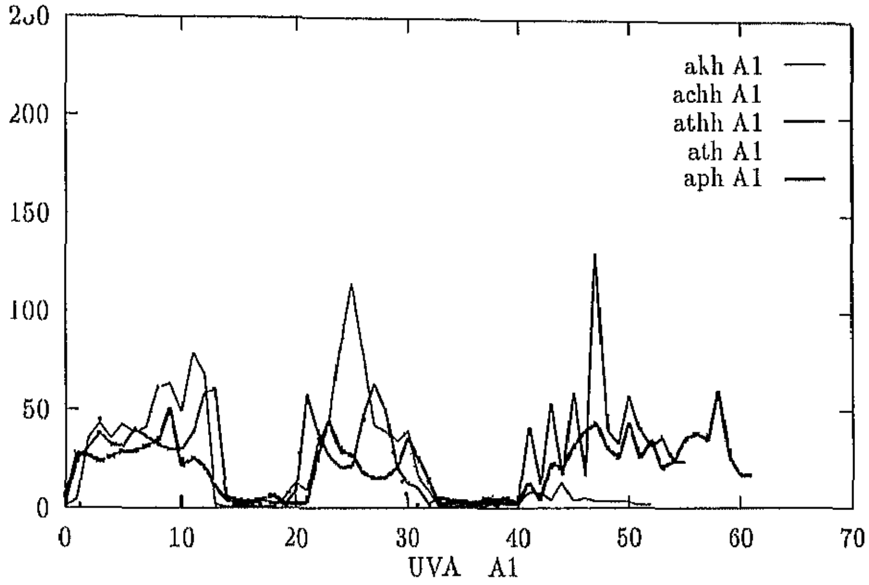


Figure 5 8 Parameter A_1 for Unvoiced Aspirated Stops

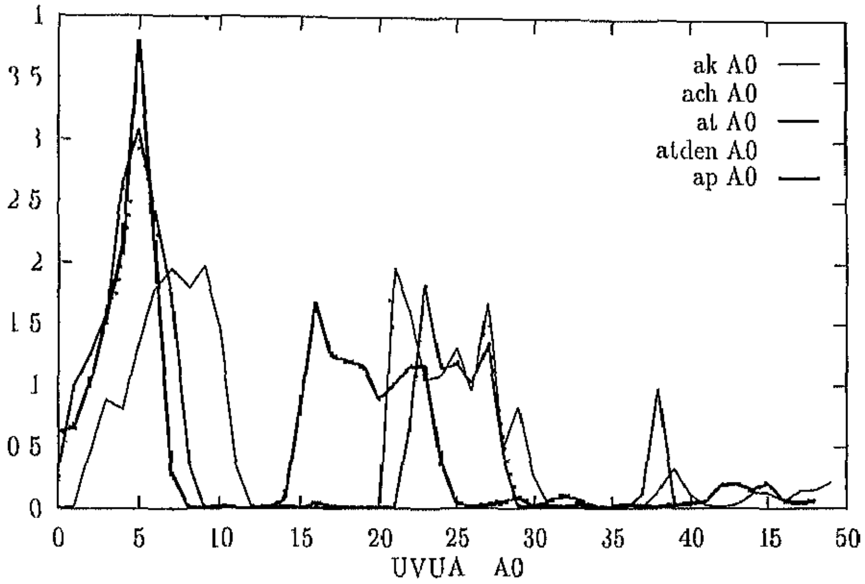


Figure 5 9 Parameter A_0 for Unvoiced Unaspirated Stops

5 4 Interpretations

These results make the following points clear

1 The area functions should be calculated with pitch synchronous analysis Since the method adopted was not pitch synchronous, we get oscillations in sustained vowel part

2 The behaviour of two classes of consonants the voiced aspirated stops and the bilabial stops is remarkable and seem to be less tractable using the present model

3 The overall constriction of vocal tract is detectable but not the 'stop' phenomenon which makes the stops 'stop's Thus our premise of getting similarity in behaviour row wise and difference in behaviour column wise fails The possible cause behind this may be the averaging out in space and time Details of this interpretation are discussed in Chapter 6

Chapter 6

Conclusion and Suggestions for Future Work

6 1 Conclusions

As is evident from the results, the waveguide model fails to detect the full constriction This may be because of the following causes

- The full constriction is too transitory The detection becomes more difficult if the frame of analysis divides the constriction interval into two parts

- The constriction is not limited to a particular section of vocal tract but the tongue movement constricts the whole vocal tract for the time being Thus averaging takes place not only in the time but across the area parameters also

- The full constriction does not constrict one whole section of the vocal tract in most of the cases Thus the phenomenon gets averaged out over the whole section

6.2 Suggestion for Future Work

- In order to be more confident about the vowel part, a pitch synchronous analysis should be carried out. This will help in co-articulation studies of utterances.

- The problem associated with a transitory phenomenon can be alleviated by choosing smaller frame of analysis. The accuracy of the analysis can be maintained by increased rate of sampling. But such a analysis will be useless for reproduction because the synthesis model assumes a set of gain,pitch and reflection coefficients fed at a time. Smaller interval of analysis shall have *error in pitch detection*. The simplest change in the model to circumvent the problem is to have multiple gain reflection coefficient analysis over the pitch period and thus several changes in vocal tract to be driven by single excitation.

- The problem of spatial distribution of construction can be tackled by increasing the number of cylindrical sections, thus increasing the number of poles in the filter. The other possible approach is to assume a waveguide with sections nonuniform in length. In either of the cases, we lose *the simplicity of the model however*.

- Thus our model may not be suitable for text to speech conversion system for an orthography based on articulatory phonetics. This problem may be overcome by incresed number of rules leading to such an implementation.

Appendix A

The recording was carried out on a female and a male speaker in form of isolated V CV utterances The care was taken not to have any contextual meaning of any utterance The recording was monophonic

The sampling frequency was set to 8000 Hz The filter cutoff frequency was set to be 3500 Hz The duration of recording was 2 S wherein the silence was arbitrarily recorded preceding and following the utterance

As per the specifications supplied by the manufacturer, the filtering is done by eighth order programmable active Butterworth lowpass filters They have a roll off of 48 dB/octave [7]

Bibliography

1 Rabiner L R , Schafer R W , "*Digital Processing of Speech Signals,*" Prentice Hall, Inc , Englewood Cliffs, New Jersey 07632, 1978

2 Markel J D , Gray A H Jr , "*Linear Prediction of Speech,*' Springer Verlag, Berlin, Heidelberg, New York, 1976

3 Witten I J , "*Principles of Computer Speech,*" Academic Press, London, 1982

4 Gersho A , "*Advances in Speech and Audio Compression,*" Proc of IEEE, vol 82, No 6, *pp* 900 918, June 1994

5 Shastri Shridharanand, "*Laghoo Siddhanta Kaumudi,*" Motilal Banarasidas, Bunglow Road, Jawahar Nagar, Delhi 7^{th} edition, 1975

6 Chaturvedi,A P Sitaram, "*Vag Vijnana,*", Choukhambha Vidyabhavan, Varanasi, 1969

7 Kelkar A P , "*Studies in Hindi Urdu,*", Part I, Deccan College, Post graduate and Research Institute,Pune, 1968

8 Sloat C , Taylor S H , Hoard J E "*Introduction to Phonology,*" Prentice Hall, Inc , Englewood Cliffs, NJ 07632, 1972

9 "*User's Manual, TSP 53C30,*" Texas Instruments, 1994

10 "*User's manual, Speech Interface Unit,*", Model 3, Voice and Speech Systems, Malleswaram, Bangalore 560 013 April 1991

11 Markel J D , "*The SIFT Algorithm for Fundamental Frequency Estimation,*" IEEE Trans on Audio and Electroacoustics, vol AU 20, No 5, *pp*367 377, Dec 1972

12 Rabiner L R , Cheng M J , Rosenberg A E , McGonegal, "*A Comparative Performance Study of Several Pitch Detection Algorithms,*" IEEE Trans Acoust ,Speech and Signal Proc , Vol ASSP 24, No 5, *pp* 399 418, October 1976